

Процесорни регистри - предназначение, структура, операции в регистрите.
Класификация на регистрите. Регистри с общо предназначение, адресни регистри, сегментни регистри, управляващи регистри. Програмен модел - пример

1. Процесорни регистри - предназначение, структура, операции в регистрите.
Класификация на регистрите. Регистри с общо предназначение, адресни регистри, сегментни регистри, управляващи регистри.

1.1. Процесорни регистри:

Процесорните регистри представляват блокът вътрешни регистри (вж.Фиг.3), който е един от съставлящите основни блокове на централния процесор. Другият основен блок, който беше разгледан в предишната тема е АЛУ.

1.2. Класификация на регистрите:

По предназначение вътрешните регистри могат да бъдат разделени на две основни групи: регистри за данни и регистри за адреси. От своя страна регистрите за адреси се делят на две подгрупи:

регистри указатели

(SP, BP, IP) и

индексни регистри

(SI, DI).

1.3. Регистри с общо предназначение, адресни регистри, сегментни регистри, управляващи регистри:

По видове различаваме следните регистри в блока вътрешни регистри:

- Регистър файл: регистър за съхранение на файл. Файл наричаме съвкупност от

Написано от sevda

Петък, 26 Април 2013 07:09 -

данни с общо предназначение и тип, която притежава три атрибута: път, име и разширение.

- Указател на стека (SP- Stack Pointer: stack-магазин), респективно ESP, представлява 16 битов регистър (SP), разширяем до 32 бита (ESP) на съвременните процесори, съдържащ адреса на следващата свободна клетка от стека. Той -показва (указва) началото на адресния стек и на стека за данни, които се намират в паметта и са необходими при работа с подпрограми. Стекът има организация „пръв влязъл-пръв излязъл”(FIFO), с достъп четене/запис. Това позволява съхранение на данните по време на прекъсванията и при изпълнение на подпрограми при тези прекъсвания. При запис на всеки нов байт в стека, SP автоматично се намалява, докато при четене SP се увеличава.

Указателят на базата (BP-Base Pointer), респективно EBP, се използва както регистър

а за данни

BX за пресмятане на адрес при специални видове адресиране, свързан с регистър за стеков сегмент (SS).

- Индексни регистри: Те включват:

Регистър за първични индекси (SI -Source Index), респ. ESI и Регистър за целеви индекси (DI -destination index),

ESI
респ.

EDI

. Тези индексни регистри служат за пресмятане на адрес при инструкции за низове, ако например низ от знаци трябва да бъде преместен вътре в паметта. Тези регистри са ориентирани по X и по Y.

SI по X

се използва за индексна адресация, като подава 16 битова индексна стойност, която се сумира с 8-битово отместване в инструкцията, за да се изчисли действителния адрес. Този регистър може да бъде използван и като брояч или регистър за временно съхранение.

SI по

Y

, подобно на този по X, се използва за индексна адресация. Неговото използване изисква един допълнителен байт в машинния код на инструкцията, тъй като всички инструкции с

SI по

Y

са с двубайтови кодове.

Указателят на инструкциите (IP -Instruction Pointer), респ. EIP, показва адресите на

Написано от sevda

Петък, 26 Април 2013 07:09 -

следващите, наредени в опашка инструкции. Това отличава този регистър от брояча на инструкции (респ. от указателя на инструкциите в по-старите микропроцесори), който брояч постоянно показва следващата инструкция за изпълнение. Само при инструкция за преход, респ. за разклонение в програмата, указателят на инструкциите показва адреса на следващата за обработка инструкция.

- Сегментни регистри: Това са два допълнителни сегментни регистъра:

Флагов сегмент (FS - Flag-Segment и

Кодов сегмент (CS - Code-Segment).

При тези допълнителни сегментни регистри, фирмата Intel е избрала следващите след „E” букви от латинската азбука. Тази фирма Intel, разработва друг начин за образуване на адреса. При този начин всеки адрес в паметта се получава чрез сумиране на един базов адрес и съответното отместване - Offset. Базовият адрес се запомня в един сегментен регистър, а отместването се намира в един от указателите или индексните регистри - например в указателя на инструкциите, ако дадена инструкция трябва да бъде прочетена от паметта. При директен достъп до данните в паметта, това отместване се задава при самата инструкция. Действието е следното: процесорът образува необходимия, примерно 20-битов адрес, с който трябва да се обърне към оперативната памет. Това е т.н. „физически адрес”, като процесорът умножава базовия адрес на 16 (=10_h), което в двоичната бройна система съответства на простото за реализация преместване с 4 бита наляво, след което прибавя отместването.

Сегментните регистри

са четири типа:

CS

-кодов сегмент

;

DS

-

сегмент за данни

;

SS

-стеков сегмент

и

ES

-

допълнителен сегмент за данни

Написано от sevda

Петък, 26 Април 2013 07:09 -

. Същественото предимство на образуването на сегменти е възможността за преместване на програмите и данните в паметта. Това свойство се нарича

преместваемост -
relocatability

.

- Регистри за данни: Това са четири 16-разрядни регистъра: AX, BX, CX и DX, респективно четирите 32-разрядни регистъра при процесорите на фирмата Intel:

EA

X

,

E

BX

,

E

CX

и

E

DX

. тези 32-разрядни регистри са предназначени преди всичко за съхраняване на данните. Тъй като често се работи с еднобайтов операнд, 16-разрядните регистри са разделени на два 8-разрядни: старши - H-Higher и младши - L-Lower, регистри. Например регистърът

AX

се състои от

AH

и

AL

. Заедно с предназначението си като регистри за данни те изпълняват още и специални функции:

ü Регистърът AX - респ. EAX, наричан също така акумулатор, е централен регистър за умножение и деление. За извеждане към входно-изходен канал се използват регистрите AL, AX или EAX;

ü Регистърът BX, наричан още базов регистър (Base Register) е необходим за пресмятане на адреса при специален вид адресиране;

Написано от sevda

Петък, 26 Април 2013 07:09 -

ÿ Регистърът CX, наричан също броячен регистър (Count Register), служи като броячен регистър при цикли, в инструкции за преместване надясно, респ. наляво, както и при повторяеми инструкции за низове. При последните може да се използва и регистърът ECX;

ÿ Регистърът DX, респ. EDX, наричан също регистър за данни (Data Register), се използва за пресмятане на адреса при специален вид адресиране. Тези регистри DX

, свързани с

AX

(респ.

E

DX

, свързани с

EA

X

), се използват при умножение, респ. при деление, както беше посочено по-горе при

AX

или

EA

X

.

- Програмен брояч(PC -Program Counter): Това е най-често 16-разряден регистър, който съдържа адреса на следващата инструкция, чието изпълнение предстои.

- Регистър за код на условията (CCR -Code Condition Register). Това е най-често 8-разряден регистър, в който всеки бит указва определено условие от резултата на току-що изпълнената инструкция. Тези битове могат да бъдат тествани индивидуално от програмата и в резултат от тези тестове могат да бъдат предприети различни действия.

- Дешифратор на инструкции: Този регистър дешифрира кода на операцията (КОП) и определя коя е заявената за изпълнение операция. Това представлява обработка на текущата инструкция.

Сега ще се спрем на един от основните въпроси в работата на КС - структура на инструкциите. За целта с обща схема ще представим цикъла на всички основни структури на инструкциите. Цикълът на една инструкция се състои от код на операцията (КОП) и адресно поле (АП) на инструкцията:

Написано от sevda
Петък, 26 Април 2013 07:09 -

КОП

=====

АП

=====

Безадресна

КОП

Едноадресна

КОП

A1

Двуадресна

КОП

A1

A2

Триадресна

КОП

A1

A2

A3

Триадресната инструкция е най-близка до традиционната операция. Тя представя двата адреса на двата операнда и адресът, където ще се съхрани резултата: $(A1) + (A2) = (A3)$.

При двадресната структура: $(A1) + (A2) \rightarrow (A1)$.

Едноадресните структури се използват при процесори, в които има служебен вътрешен регистър, наречен акумулатор (A_k): $(A_k) + (A1) \rightarrow (A_k)$.

Безадресната инструкция се появява тогава, когато има нулиране на служебен регистър или установяване на даден флаг в избрано състояние.

И накрая на тази точка следва да се дефинира размера на думата в КС: тя е 8-разрядна, което е 1 байт или най-често думата в съвременните КС е двубайтова (2 байта).

1. Програмен модел - пример.

Програмният модел на всяка компютърна система -КС (или компютърна конфигурация) е изграден от регистрите на нейния централен процесор (ЦП - CPU). С други думи - програмният модел обединява програмно достъпните регистри от структурата на КС. Техният състав и предназначение са различни за различните модели процесори, а дължината им (тяхната разрядност) зависи от дължината на машинната дума - W , за конкретната КС.

Акуму

латори

Индексни регистри

Указател на стека

Програмен брояч

Регистър за код на условията

Написано от sevda

Петък, 26 Април 2013 07:09 -
